

Jul 21

1. A multiple processor computer system comprising:
- a plurality of processors;
 - a shared resource;
 - a main memory, at least a portion thereof comprising a control structure for controlling a lock on said shared resource; and
 - a crossbar structure for controlling access among the processors to the shared resource, the crossbar structure comprising, for each processor, a corresponding storage location, one of the plurality of processors writing to a storage location corresponding to the one processor, an address of the lock control structure associated with said shared resource to acquire the lock thereto, the crossbar structure, on behalf of the one processor, performing memory operations on the lock control structure at the address specified in the corresponding storage location in order to acquire the lock on behalf of the one processor.
2. The multiple processor computer system of claim 1 wherein the crossbar structure further comprises a second storage location corresponding to a respective processor in the multi-processor system, one of the plurality of processors writing to the corresponding second storage location an address of the lock control structure associated with said shared resource to release the lock thereto, the crossbar structure, on behalf of the processor, performing memory operations on the lock control structure at the address specified in the corresponding second storage location in order to release the lock on behalf of the processor.
3. The multiple processor computer system of claim 2 wherein the crossbar structure further comprises a third storage location corresponding to a respective processor in the multi-processor system for storing a current interrupt priority level associated with the processor wherein the current interrupt priority level represents the level of interrupt that the processor will service while spinning on a lock.
4. The multiple processor computer system of claim 3 wherein the crossbar structure further comprises a fourth storage location corresponding to a respective

processor in the multi-processor system for storing a future interrupt priority level associated with the processor wherein the future interrupt priority level represents the level of interrupt that the processor will operate at upon receiving a lock.

5. The multiple processor computer system of claim 1 wherein the crossbar structure further comprises a queue for determining which processor is granted a lock.

6. The multiple processor computer system of claim 5 wherein the queue comprises a first data field for containing a processor identification, a second data field for containing a lock request time, and a third data field for containing a priority of a task requesting a lock.

7. The multiple processor computer system of claim 6 wherein the queue determines which processor is granted a lock based on the lock request time and the priority of the task requesting the lock.

8. The multiple processor computer system of claim 1 further comprising a multi-processor module.

9. The multiple processor computer system of claim 8 wherein the multi-processor module comprises/at least two of the plurality of processors.

10. The multiple processor computer system of claim 1 wherein the shared resource comprises at least a portion of memory.

11. The multiple processor computer system of claim 1 wherein the shared resource comprises an I/O device.

12. / The multiple processor computer system of claim 1 wherein the shared resource comprises a register.

for each processor, a corresponding storage location for receiving from the respective processor a memory address of a lock control structure associated with said shared resource;

14. The crossbar structure of claim 13 further comprising:

wherein to release a lock thereto, the crossbar structure, on behalf of a processor, performs memory operations on the lock control structure at the address specified in the corresponding second storage location in order to release the lock on behalf of the processor.

for each processor, a corresponding third storage location for receiving from the respective processor a current interrupt priority level associated with the processor wherein the current interrupt priority level represents the level of interrupt that the processor will service while spinning on a lock.

for each processor, a corresponding fourth storage location for storing a future interrupt priority level associated with the processor wherein the future interrupt priority level represents the level of interrupt that the processor will operate upon receiving a lock

5/13/17

18. The crossbar structure of claim 17 wherein the queue comprises a first data field for containing a processor identification, a second data field for containing a lock request time, and a third data field for containing a priority of a task requesting a lock.

[illegible]

25. The method of claim 24 wherein the determining a processor further comprises:

10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532

reading at least one priority and a processor identification corresponding to each priority, from the queue;

selecting the highest priority of the read at least one priority;

determining a processor based on the processor identification corresponding the selected highest priority; and

granting the lock to the determined processor corresponding to the selected highest priority.

26. The method of claim 23 wherein storing the future interrupt priority level further comprises copying a value contained in a current interrupt priority register to a future interrupt priority register prior to writing, by one of said processors, the memory address.

27. The method of claim 26 further comprising copying a value contained in a future interrupt priority register to a current interrupt priority register.

09851795:050901

34. The method of claim 28 further comprising receiving a current interrupt priority level associated with said one processor wherein the current interrupt priority level represents the level of interrupt that said one processor will service while spinning on a lock.

[illegible]

36. The method of claim 28 further comprising determining a processor from the plurality of processors to be granted a lock based on a queue.

reading at least one priority and a processor identification corresponding to each priority, from the queue;

determining a processor based on the processor identification corresponding to the selected highest priority; and

granting the lock to the determined processor corresponding to the selected highest priority.

38. A computer-readable medium having instructions stored thereon for use in a multi-processor computer system having a crossbar structure that connects a plurality of processors to at least one shared resource, the instructions, when executed on the crossbar structure, causing the crossbar structure to perform the following:

receiving at the crossbar from one of said processors, a memory address of a lock control structure associated with said shared resource when the processor needs to acquire a lock thereto; and

performing at the crossbar structure, on behalf of said one processor, a memory operation on the lock control structure at the memory address received from said one processor in order to acquire the lock on behalf of said one processor.

39. The computer-readable medium of claim 38 wherein the instructions further cause the crossbar to perform storing the memory address received from said one processor in a respective storage location within the crossbar structure.

40. The computer-readable medium of claim 38 wherein the instructions further cause the crossbar to perform determining a status of the lock control structure at the memory address received from said one processor.

41. The computer-readable medium of claim 40 wherein the instructions further cause the crossbar to perform sending to said one processor the determined status of the lock control structure.

42. The computer-readable medium of claim 38 wherein the instructions further cause the crossbar to perform performing at the crossbar structure memory operations on the lock control structure at the memory address received from said one processor, continually, until an unlocked status is received.

43. The computer-readable medium of claim 38 wherein performing a memory operation comprises performing a test and set operation.

00851705 : 050001
T06050

44. The computer-readable medium of claim 38 wherein the instructions further cause the crossbar to perform receiving a current interrupt priority level associated with said one processor wherein the current interrupt priority level represents the level of interrupt that said one processor will service while spinning on a lock.

45. The computer-readable medium of claim 38 wherein the instructions further cause the crossbar to perform storing a future interrupt priority level associated with said one processor wherein the future interrupt priority level represents the level of interrupt that said one processor will service upon receiving a lock.

46. The computer-readable medium of claim 38 wherein the instructions further cause the crossbar to perform determining a processor from the plurality of processors to be granted a lock based on a queue.

47. The computer-readable medium of claim 46 wherein the determining a processor further comprises:

reading at least one priority and a processor identification corresponding to each priority, from the queue;

selecting the highest priority of the read at least one priority;

determining a processor based on the processor identification corresponding to the selected highest priority; and

granting the lock to the determined processor corresponding to the selected highest priority.

09851705:050001

ADDALY